International Journal of Enhanced Research in Management & Computer Applications
ISSN: 2319-7471, Vol. 13 Issue 8, August, 2024, Impact Factor: 8.285

Steel Defect Detection using Machine Learning

G. Arunalatha

Assistant Professor, Department of Computer Science and Engineering, Perunthalaivar Kamarajar Institute of Engineering and Technology (PKIET), Karaikal. Puducherry

ABSTRACT

The steel plate stands as a primary output in the steel sector, and the quality the steel plays a crucial role in determining the performance of the final product. Identifying surface flaws in steel plates while they are being produced is a difficult task. Utilizing machine learning algorithms to forecast the quality of products made in the industry is necessary. These algorithms are being increasingly applied to enhance defect detection in manufacturing by automatically categorizing data. The approach for detecting surface flaws through machine vision is gaining popularity due to its high precision, rapid speed, and smart analysis capabilities, marking a significant trend in this area. The proposed work uses LTP for technique for defect detection in steel plates. This work presents an investigation of various machine learning algorithms to defect classification in steel plate manufacturing using the NEU Surface Defect Database dataset.

Keywords: Fault detection, Steel plates, Machine learning

INTRODUCTION

The most common type of metal is iron. This metal, which finds application in many different fields, undergoes various processing steps to become steel. Because steel is more brittle than iron, during the rolling process, flaws could appear in the plates. In terms of both safety and commerce, it is critical that these flaws are found during production. Steel plates are utilized in the society in various areas such as the building of bridges, automobiles, aircraft, etc. The dependability of public use, society's safety, and public industry safety are all directly correlated with the quality of the metal plate. The steel rolling process will result in a range of defects because of various factors such as the production process, the environment, and other constraints. A key factor in raising the quality of steel, production efficiency, and related smart manufacturing is the intelligent monitoring and diagnosis of steel defects. In mechanical engineering, the use of artificial intelligence and bioinspired computation is progressively becoming more significant. Defect detection in manufacturing is usually done manually to guarantee the necessary quality of the produced parts; but because it takes a lot of time and labor, a lot of research focuses on automating this process. In these situations, machine learning techniques can be applied to provide quick and highly accurate detection.

LITERATURE SURVEY

An approach using Generative Adversarial Networks [1] is suggested for creating artificial training images. A new design is created with a type-specific variable in the Generator and a classification section included in the Discriminator. Using a larger dataset, two detection algorithms, Faster R-CNN and YOLO, are utilized.

A better ELM algorithm [2] paired with a Genetic Algorithm was suggested and utilized to detect surface defects on hot rolled steel plates. The output matrix of the hidden layers in ELM was viewed as a chromosome, with new iteration rules incorporated. The algorithm underwent testing with 1675 hot rolled steel plate samples, which contained various imperfections like pockmarks, chaps, scars, longitudinal cracks, scratches, scales, transverse cracks, scratches, and roll marks. The outcomes indicated that the G-ELM (Genetic Extreme Learning Machine) algorithm achieved the highest identification accuracies of 98.46% for the training set and 94.30% for the testing set.

TheDF-ResNeSt50 network model, presented in [3], aims to study steel flaws by analyzing surface defect data from steel plates. The data undergo preprocessing steps like ColorJitter, Random VerticalFlip, Normalize, and more. The model integrates feature pyramid networks and split attention network inspired by the visual attention mechanism in the bionic algorithm. The network is enhanced and optimized through data improvement, fusion of multi-scale features, and optimization of the network structure. This model presents a basic split attention block that is modularized, aiming to enhance the attention mechanism of cross-feature graph groups.

A method for detecting periodic defects was designed using a convolutional neural network (CNN) and long short-term memory (LSTM) [4] to match the time-sequenced nature of these defects. Extracting defect image features is done via a CNN network, followed by feeding the feature vectors into an LSTM network for defect identification.

A system for inspecting defects[5] is suggested, which employs a double lighting setup to differentiate between uneven defects and color variations caused by surface irregularities. An algorithm is presented for image processing that is used to identify defects. The Gabor filter algorithm identifies the switching pattern and uses binarization to extract the defect's shape. The optics module and detection algorithm, which were improved using a simulator, were implemented in an actual facility.

An approach using machine vision [6] is used to detect and classify surface defects commonly found in used gun barrels, including normal wear, corrosive pitting, rust, and erosion. A CCD camera with a small microscopic probe was utilized to non-destructively capture images of the faulty areas on multiple used gun barrels. Normal wear was depicted as bright among the captured images, while the other three defects were shown as dark. Various textural characteristics were obtained from the segmented images using histogram and gray level co-occurrence matrix, and were then ranked automatically using the sequential forward feature selection method to choose the most optimal features for classification. Several classifiers, including those utilizing Bayes, k-Nearest Neighbor, Artificial Neural Network, and Support Vector Machine (SVM), were evaluated, with the findings showing that SVM was the most effective for this particular use case.

CNNs were trained on a dataset of photometric stereo images showing metal surface defects, specifically rail defects [7]. These flaws are holes on the rail's surface and signal potential surface damage leading to rail breakage. CNNs were trained with supervision and additional methods like unsupervised pre-training and data augmentation were studied.

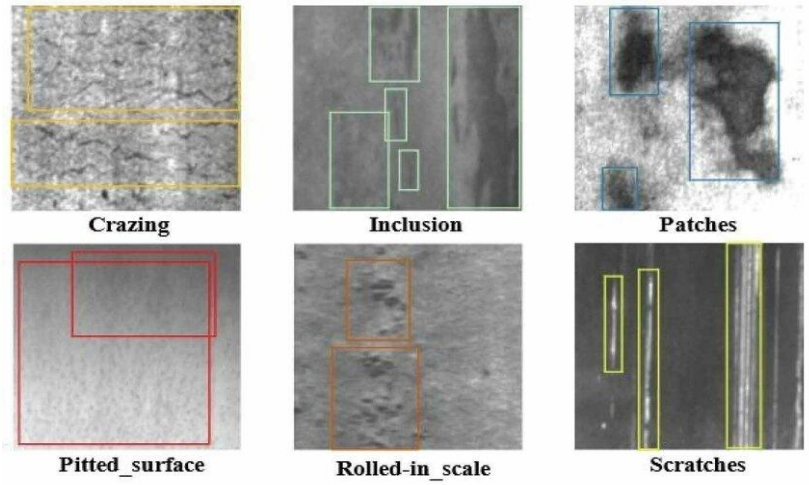

Fig:1 Types of Surface Defects

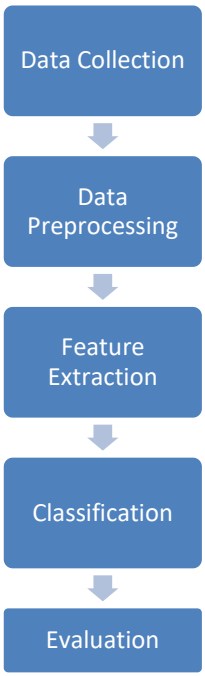

Fig:2 Surface Defect Detection Model

PROPOSED WORK

NEU Surface Defect Database

In NEU Surface Defect Database [8], Six kinds of typical surface defects of the hot-rolled steel strip are collected, i.e., rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc). The database includes 1,800 grayscale images: 300(split into 240 images for training and 60 images for testing.) samples each of six different kinds of typical surface defects

Data preprocessing

Data preprocessing is the process of evaluating, filtering, manipulating, and encoding data so that a machine learning algorithm can understand it and use the resulting output. A Mean Filter is a smoothing filter in computer science that calculates the average brightness of a pixel and its neighboring pixels to replace the original brightness, thus reducing noise in an image. The major goal of data preprocessing is to eliminate data issues such as missing values, mean filter is one of the techniques which is used to reduce noise of the images. This is a local averaging operation and it is a one of the simplest linear filter. The value of each pixel is replaced by the average of all the values in the local neighborhood. Let f(i,j) is a noisy image then the smoothed image g(x,y) can be obtained by,

$$g(x,y)= \frac{1}{n} \sum_{(i,j) \in S} f(i,j)$$

Where S is a neighborhood of (x,y) and n is the number of pixels in S.

Feature Extraction using Local Ternary Pattern

The primary focus of LBP is on the texture characteristics present in the image. Typically, a block with dimensions of 3 × 3 is selected. The middle pixel is compared to all surrounding pixels. Values are given to neighboring pixels according to how they compare with the central pixel. If the value of the central pixel is greater than a neighboring pixel, it will be given a value of 1; if not, it will be given a value of 0. After setting the binary values, the binary string is created by examining all the neighboring values either in a clockwise or anticlockwise direction. Next, the binary string is transformed into its decimal equivalent, representing the new value of the pixel. A histogram is created by tallying the frequency of each number. This feature vector is represented by the histogram. The LBP can be affected by noise, as even a slight change in gray value of the central pixel can result in varied codes for a neighborhood in an image, particularly in smooth areas. To address this weakness, LTP was developed as an expansion of the original LBP, creating a version with three-value codes known as the local ternary pattern (LTP).

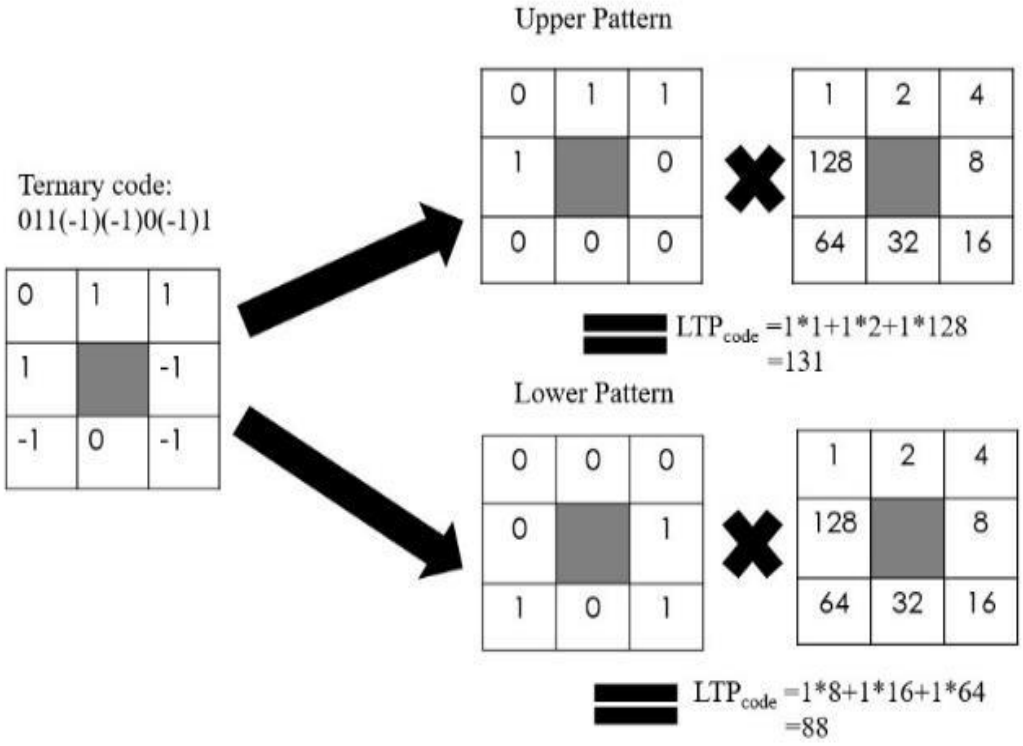

Fig:3 LTP Operation

The LTP operator is an extension of the LBP descriptor that assumes up to 3 coded values ({−1, 0, 1}). This is achieved by changing the step function S in the following manner: Considering k as the threshold constant, c as the value of the center pixel, a neighboring pixel p, the result of threshold is:

$$\text{LTP(i)} = \begin{cases} 1, \text{if } p > c + k \\ 0, \text{if } p > c - k \text{ and } p < c + k \\ -1, \text{if } p < c - k \end{cases}$$

We split the LTP code into two LBP codes (with only positive values). First, we create the upper pattern by converting the negative codes to zero. Next, we create the lower pattern by setting the positives values to zero, by converting the negative values to positive.

We calculate the LTP value of each block. We convert each value of a block Lj into ternary bits as follows. The center element is referred to as Ec and neighbors (other elements) are referred to as Ei , where i indicates the index of the element. Ternary values are then assigned to create the Ternary string. If (Ec − Ei) is greater than or equal to a threshold then we assign a −1 to ith element. If (Ei − Ec) is greater than or equal to a threshold, we assign it as +1. For all other cases, we assign 0.

CLASSIFICATION ALGORITHMS

Logistic Regression
Logistic regression [10] is a statistical technique employed in the development of machine learning models where the dependent variable is binary in nature. This method is utilized to analyze data and the relationship between a single dependent variable and one or more independent variables, which may be nominal, ordinal, or interval in type. In the context of logistic regression, the objective is to predict the likelihood that a specific input belongs to a particular class.

This involves estimating the parameters of a logistic function that correlates the input features with the probability of the output class. The primary aim is to identify the model that best fits the data by maximizing the likelihood of the observed outcomes. Logistic regression utilizes the logistic function, also referred to as the sigmoid function, to represent the relationship between input features and the probabilities of the output classes. This function effectively maps any real number to a range between 0 and 1, making it ideal for probability modeling. The parameters of the model are determined through optimization techniques such as gradient descent, which incrementally adjusts the parameters to reduce the discrepancy between predicted probabilities and actual class labels.

Decision Tree
A decision tree [9] is a structured representation resembling a flowchart, where each internal node signifies a feature (or attribute), branches denote decision rules, and leaf nodes indicate outcomes. The root node is the topmost element of the decision tree. This model learns to partition data based on the values of attributes, employing a recursive technique known as recursive partitioning. In the realm of machine learning, decision trees serve as a powerful tool for decision-making, as they clearly outline the problem along with all potential outcomes. This structure allows developers to evaluate the implications of various decisions, and as the algorithm processes additional data, it enhances its ability to forecast outcomes for new data sets.

Random Forest
The Random Forest Algorithm [10] is a widely utilized supervised machine learning technique that addresses both classification and regression challenges. Analogous to a forest consisting of numerous trees, the robustness of the algorithm increases with the number of trees employed. Consequently, a higher quantity of trees within a Random Forest enhances its accuracy and problem-solving capabilities. This algorithm functions as a classifier by aggregating multiple decision trees derived from various subsets of the dataset, ultimately averaging their outputs to bolster predictive accuracy. It operates on the principle of ensemble learning, which involves the integration of multiple classifiers to tackle complex problems and enhance model performance.

Support Vector Machine (SVM)
A Support Vector Machine (SVM) [9] is a supervised learning algorithm employed in machine learning for addressing classification and regression tasks. SVMs excel particularly in binary classification scenarios, where the objective is to categorize data points into two distinct groups. The primary goal of an SVM is to identify the optimal line, or decision boundary, that effectively separates the data points belonging to different classes. In high-dimensional feature spaces, this boundary is referred to as a hyperplane. The methodology focuses on maximizing the margin, defined as the distance between the hyperplane and the nearest data points from each class, thereby facilitating the clear distinction between data categories.

Naive Bayes

The Naive Bayes [9] Algorithm is a fundamental technique in machine learning, particularly effective for addressing classification challenges. It is based on Bayes' theorem of probability and is particularly useful in text classification tasks, where it processes high-dimensional datasets. Notable applications of the Naive Bayes Algorithm include sentiment analysis, categorization of news articles, and spam detection. The foundation of this algorithm lies in probability, as it generates probabilistic outcomes that assist in resolving complex problems through predictive analysis.

K-Nearest Neighbors (KNN)

The k-nearest neighbors (KNN) [9] algorithm is a non-parametric, supervised learning method that classifies data points based on their proximity to other data points. This approach operates on the principle of identifying a specified number of training samples that are closest to a new data point and predicting its classification based on these samples. The number of neighbors can be a fixed value determined by the user (k-nearest neighbor learning) or can adjust according to the local density of data points (radius-based neighbor learning). Various distance metrics can be employed, with the standard Euclidean distance being the most frequently utilized.

EVALUATION METRICS

i) Accuracy

Accuracy is a fundamental evaluation metric for assessing the overall performance of a classification model. It is the ratio of the correctly predicted instances to the total instances in the dataset.

Accuracy=TP+TN/TP+FP+TN+FN

Precision

Precision evaluates the accuracy of the positive prediction made by the classifier. In precision specifieshow many were actually positive in the total number of instances.

Precision = (TP) / (TP) + (FP)

Recall

The recall is also known as sensitivity or true positive rate. It is the ratio of the number of true positive predictions to the total number of actual positive instances in the dataset. Recall measures the ability of a model to identify all relevant instances.

Recall = (TP) / (TP) + (FN)

F1-Score

F1 score is the harmonic mean of precision and recall. It provide a single metric that balances the trade-off between precision and recall. It is especially useful when the class distribution is imbalanced.

F1 Score = 2 x [(Precision x Recall)/ (Precision + Recall)]

Table.1 Accuracy of Classifiers

Classifiers	Accuracy
Logistic Regression	83.5%
Decision Tree	85.2%
Random Forest	84.5%
Support Vector Machine	77.6%
, Naïve Bayes	78.3%
K Nearest neighbors	83%

CONCLUSION

Identifying defects in steel plates is essential for maintaining the safety and dependability of buildings and machinery. Spotting these defects early on can stop additional harm and expensive fixes. Surface defects on steel plates significantly impact their quality. Therefore, it's vital to identify these defects using online systems for surface inspection. Steel plate defect detection can be done using machine learning techniques. In this work, features are extracted from the steel plate image of NEU Surface Defect Database using Local Ternary Pattern(LTP) method and

Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Naïve Bayes and K Nearest neighbors machine learning techniques were applied for classification of steel plate images as good and fault.

REFERENCES

[1]. Renjie, Tang & Mao, Keming. (2020). An Improved GANs Model for Steel Plate Defect Detection. IOP Conference Series: Materials Science and Engineering. 790. 012110. 10.1088/1757-899X/790/1/012110.
[2]. Tian, S.; Xu, K. An Algorithm for Surface Defect Identification of Steel Plates Based on Genetic Algorithm and Extreme Learning Machine. Metals 2017, 7, 311.
[3]. Hao, Zhiqiang& Wang, Zhigang& Bai, Dongxu& Tao, Bo & Tong, Xiliang& Chen, Baojia. (2022). Intelligent Detection of Steel Defects Based on Improved Split Attention Networks. Frontiers in Bioengineering and Biotechnology. 9. 10.3389/fbioe.2021.810876.
[4]. Liu, Y.; Xu, K.; Xu, J. Periodic Surface Defect Detection in Steel Plates Based on Deep Learning. Appl. Sci. 2019, 9, 3127.
[5]. Yun, Jong & Kim, Dongseob& Kim, KyuHwan& Lee, Sang-Jun & Park, Chang & Kim, Sang Woo. (2017). Vision-based surface defect inspection for thick steel plates. Optical Engineering. 56. 053108. 10.1117/1.OE.56.5.053108.
[6]. Shanmugamani, R.; Sadique, M.; Ramamoorthy, B. Detection and classification of surface defects of gun barrels using computer vision and machine learning. Measurement 2015, 60, 222–230
[7]. Soukup, Daniel & Huber-Mörk, Reinhold. (2014). Convolutional Neural Networks for Steel Surface Defect Detection from Photometric Stereo Images. 10.1007/978-3-319-14249-4_64.
[8]. Song, K., Yan, Y.: A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. Applied Surface Science 285, 858–864 (2013)
[9]. Kotsiantis, Sotiris &Zaharakis, I. &Pintelas, P.. (2006). Machine learning: A review of classification and combining techniques. Artificial Intelligence Review. 26. 159-190. 10.1007/s10462-007-9052-3.
[10]. Couronné, R., Probst, P. &Boulesteix, AL. Random forest versus logistic regression: a large-scale benchmark experiment. BMC Bioinformatics 19, 270 (2018). https://doi.org/10.1186/s12859-018-2264-5